Scalable Linear Solvers

University of California
Lawrence Livermore National Laboratory

Center for Applied Scientific Computing

Objective

The CASC scalable linear solvers project is developing scalable algorithms and software for the solution of large, sparse linear systems of equations on massively parallel computers having upwards of 10,000 processors.

Applications

We wish to significantly accelerate the solution of the linear systems that arise in many large-scale scientific simulation codes. Applications of interest include radiation diffusion and transport, structural dynamics, flow in porous media, and magnetic fusion energy. The linear systems result from discretizations of partial differential equations on structured, block-structured, and unstructured meshes.

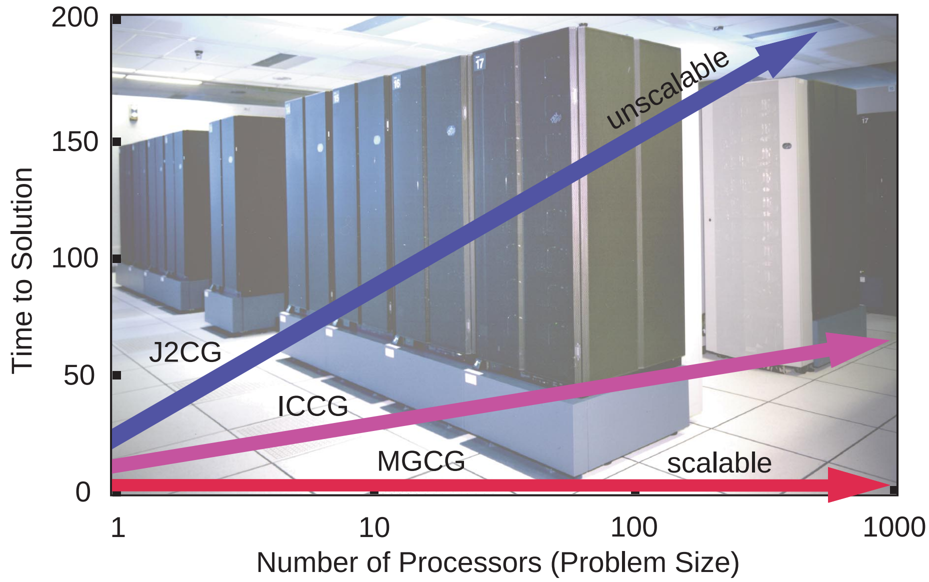

Figure 1. Scalable linear solvers (such as multigrid) enable terascale simulation by keeping solution time constant as the problem size increases with the number of processors. J2CG, ICCG, and MGCG are conjugate gradient algorithms with Jacobi, incomplete Cholesky, and multigrid preconditioners, respectively.

The Center for Applied Scientific Computing at Lawrence Livermore National Laboratory (LLNL) is developing scalable algorithms and software for solving large, sparse linear systems of equations on parallel computers. The problems of interest arise in the simulations codes being developed to study physical phenomena in the defense, environmental, energy, and biological sciences.

The Need for Scalable Algorithms

Computer simulations play an increasingly important role in scientific investigations, supplementing (and in some cases, supplanting) traditional experiments. In engineering applications, such as automotive crash studies, numerical simulation is much less expensive than experimentation. In other applications, such as global climate change, experiments are impractical (or unwise), and simulations are used to explore the fundamental scientific issues.

Finally, in the area of nuclear weapons stockpile stewardship, full-blown experiments are prohibited by the Comprehensive Test Ban Treaty, and detailed numerical simulations are needed to fill the resulting void. To address this need, the Department of Energy launched the ambitious Accelerated Strategic Computing Initiative (ASCI) project, the goal of which is to build a simulation capability to help ensure the reliability and safety of the nation's nuclear deterrence. Toward this end, codes are being developed to solve highly resolved three-dimensional problems that require the computational speed and large memory of the massively parallel ASCI computers.

Although parallel processing is necessary for the numerical solution of these problems, alone it is not sufficient; one also needs *scalable* numerical algorithms. By "scalable" we generally mean the ability to use additional computational resources *effectively* to solve increasingly larger problems. Many factors contribute to scalability, including the architecture of the parallel computer and the parallel implementation of the algorithm. However, one important issue is often overlooked: the scalability of the algorithm itself. Here, scalability is a description of how the total computational work requirements grow with problem size, which can be discussed independent of the computing platform.

Many of the algorithms used in today's simulation codes are based on yesterday's unscalable technology. This means that the work required to solve increasingly larger problems grows much faster than linearly (the optimal rate). The use of scalable algorithms can decrease simulation times by several orders of magnitude, thus reducing a two-day run on an MPP to 30 minutes (see Figure 1). Furthermore, the codes that use this technology are limited only by the size of the machine's memory because they are able to effectively exploit additional computer resources to solve huge problems.

Scalable algorithms enable the application scientist to both pose and answer new questions. For example, if a given simulation (with a particular resolution) takes several days to run, and a refined (i.e., more accurate) model would take much longer, the application scientist may forego the larger, higher fidelity simulation. He or she also may be forced to narrow the scope of a parameter study because each run takes too long. By decreasing the execution time, a scalable algorithm allows the scientist to do more simulations at higher resolutions.

Linear Solver Research Directions

In many large-scale scientific simulation codes, the majority of the run time is spent in a linear solver. For this reason, much of the scalable algorithms research and development is aimed at solving these large, sparse linear systems of equations on parallel computers.

Multigrid is an example of scalable linear solver technology. It uses a relaxation method like Gauss-Seidel to efficiently damp high-frequency error, leaving only low-frequency, or *smooth*, error. The multigrid idea is to recognize that this low-frequency error can be accurately and efficiently solved for on a coarser (i.e., smaller) grid. Recursive application of this idea to each consecutive system of coarse-grid equations leads to a multigrid V-cycle (Figure 2). If the components of the V-cycle are defined properly, the result is a method that uniformly damps all error frequencies with a computational cost that depends only linearly on the problem size. In other words, multigrid algorithms are scalable.

Our work uses two basic multigrid approaches: geometric and algebraic. For linear systems defined on structured meshes (e.g., logically rectangular meshes) and semi-structured meshes (e.g., locally refined meshes), we are developing geometric multigrid methods. An algorithm of this type was used in a three-dimensional parallel groundwater simulation (using eight million spatial zones) to speed up the linear solves by a factor of 120 with nearly 90% scaled efficiency on 256 processors of the Cray T3D. More recently, we implemented a similar algorithm in one of the ASCI performance codes. Preliminary results demonstrate the algorithmic scalability of multigrid in this multi-physics code; the linear algebra was sped up by a factor of 27, and overall simulation time was reduced 10-fold for a two-dimensional test problem (128,000 spatial zones). We have also demonstrated the scalability of this multigrid solver on the ASCI platforms. To date, the largest linear system we have solved had 64 million spatial zones, and the solution took 63 seconds on 1000 processors of ASCI-Red.

For linear systems defined on unstructured meshes, it is difficult to use geometric information in a way that is simple, straightforward, and portable from application to application. For this reason, we are developing new algebraic multigrid (AMG) methods. We are developing a parallel AMG solver to address the open research question of how to coarsen an unstructured grid in parallel. Research is also focused on improving the performance of AMG on finite element problems.

To enhance multigrid's robustness, we often use it as a preconditioner for Krylov methods such as conjugate gradients, but multigrid algorithms still tend to be somewhat problem-specific. To extend our capability to solve a wider variety of linear systems, we are also developing more general-purpose matrix preconditioners, including incomplete factorizations and sparse approximate inverses. Although these methods are typically not as algorithmically scalable as multigrid methods, good scalable implementations can show appreciable benefit on large numbers of processors. In addition, one of our research directions is to improve the algorithmic scalability of these methods by incorporating multigrid (or multilevel) techniques.

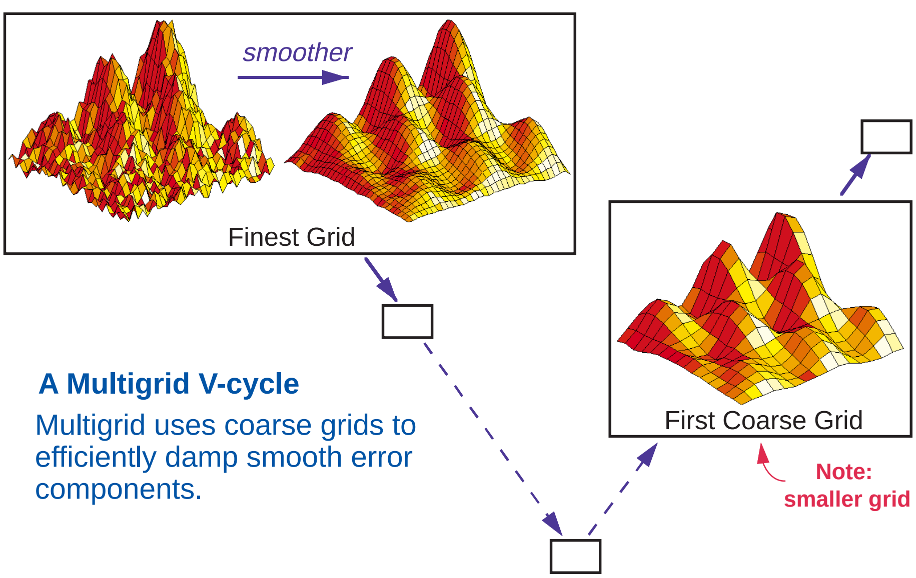

Figure 2. The down-cycle of a multigrid V-cycle uses smoothers to damp oscillatory error components at different grid scales. The up-cycle corrects the smooth error components remaining on each grid level by using the error approximations on coarser (i.e., smaller) grids.

For more information about Scalable Linear Solvers, contact Robert Falgout, (925) 422-4377, rfalgout@llnl.gov.

Work performed under the auspices of the U.S. DOE by LLNL under contract No. W-7405-ENG-48.